2019 網際網路程式設計全國大賽
高中組初賽

- 本次比賽共 7 題，含本封面共 20 頁。
- 全部題目的輸入都來自**標準輸入**。輸入中可能包含多組輸入，以題目敘述為主。
- 全部題目的輸出皆輸出到螢幕 **(標準輸出)**。
  輸出和裁判的答案必須完全一致，英文字母大小寫不同或有多餘字元皆視為答題錯誤。
- 比賽中上傳之程式碼，使用 C 語言請用 `.c` 為副檔名；使用 C++ 語言則用 `.cpp` 為副檔名。
- 使用 `cin` 輸入速度遠慢於 `scanf` 輸入，若使用需自行承擔 Time Limit Exceeded 的風險。
- 部分題目有浮點數輸出，會採容許部分誤差的方式進行評測。一般來說「相對或絕對誤差不超過 ϵ 皆視為正確」，ϵ 值以題目敘述為主。

  舉例來說，假設 $\epsilon = 10^{-6}$ 且 a 是正確答案，b 是你的答案，如果符合 $\frac{|a-b|}{\max(|a|,|b|,1)} \leq 10^{-6}$，就會被評測程式視為正確。

	題目名稱	時間限制（秒）
題目 A	珠寶	1
題目 B	貓貓與迷宮	2
題目 C	垃圾處理	1
題目 D	分裁判問題	7
題目 E	選裁判問題	1
題目 F	地底探險	2
題目 G	盤子分類	1

2019 網際網路程式設計全國大賽
輸入輸出範例

C 程式範例：

```c
#include <stdio.h>
int main()
{
    int cases;
    scanf("%d", &cases);
    for (int i = 0; i < cases; ++i)
    {
        long long a, b;
        scanf("%lld %lld", &a, &b);
        printf("%lld\n", a + b);
    }
    return 0;
}
```

C++ 程式範例：

```cpp
#include <iostream>
int main()
{
    int cases;
    std::cin >> cases;
    for (int i = 0; i < cases; ++i)
    {
        long long a, b;
        std::cin >> a >> b;
        std::cout << a + b << std::endl;
    }
    return 0;
}
```

A. 珠寶

Problem ID: jewelry

bb 是個寶石蒐藏家，某天他家旁邊新開了一間珠寶行，他希望能在這間珠寶行裡蒐集到**最多種類**的寶石。

這個世界上一共有 M 種寶石，這間珠寶行販售的 N 個珠寶組合中每個都包含了其中一些種類的寶石。bb 可以購買不限數量的珠寶組合，每個組合一旦購買了就會得到那個組合裡的所有寶石，**不能只把一部分的寶石拿走**。不過由於一些特殊的魔法，如果 bb 同時蒐集到了所有種類的寶石就會爆炸，所以他**購買的所有組合聯集起來不能含有所有種類的寶石**。

這間珠寶行有時會修改組合裡含有的寶石種類來配合消費者的喜好，每次的修改會把某種寶石加入某個組合中、或是把某種寶石從某個組合中移除。

bb 已經事先知道了珠寶行接下來**依序** Q 次的修改方式，現在他想知道如果在最一開始、以及每次的修改之後去店裡消費，**在不爆炸的前提下**，分別最多可以買到多少種寶石。

Input

輸入的第一行有三個正整數 N, M, Q 代表販售的組合個數、寶石種類數、以及珠寶行修改的次數。

接下來 N 行，每行有一個長度為 M 的 01 字串，如果第 i 行的第 j 個字元是 1，代表最一開始第 i 個組合裡含有第 j 種寶石，若是 0 則代表沒有。

接下來 Q 行，代表 Q 次的修改。每行有兩個正整數 p, t，代表改變第 p 個組合中第 t 種寶石的存在性。也就是如果該種寶石本來存在，就把他移除，不然就把他加入。

- $1 \leq N, M \leq 300$
- $1 \leq Q \leq 50000$
- $1 \leq p \leq N$
- $1 \leq t \leq M$

Output

輸出有 $Q+1$ 行，分別代表最一開始、以及每次修改後的答案。

Sample Input 1	Sample Output 1
3 4 6 1010 0000 0101 2 3 3 2 1 1 3 1 3 2 3 3	2 3 3 2 3 3 1

B. 貓貓與迷宮

Problem ID: maze

殿壬是個天才兒童，他在一個月大的時候就學會數數、六個月大的時候就學會乘法跟除法、一歲時學會寫程式、一歲又六個月時養了可愛的拉不拉多、一歲又十個月時養了可愛的貓咪、兩歲時發明了「吃餅乾」的遊戲，現在要講的是殿壬三歲大的故事。

這天，殿壬牽著他的貓貓來到了 NPSC 魔法學院的魔幻迷宮中。既然是貓貓，當然很喜歡玩躲貓貓囉！因此到了迷宮之後，貓貓就迅速地跑去躲起來了，讓殿壬來找貓貓！

魔幻迷宮可以用一個高度為 N，寬度為 M 的方格圖表示，左上方的座標是 $(1, 1)$，右下方的座標是 (N, M)，其中往下為 $+x$ 方向，往右為 $+y$ 方向。迷宮中有些格子是無法穿越的障礙物，剩下的格子則可以自由通行。

這個迷宮之所以稱作魔幻迷宮，是因為每次迷宮中的一個格子可能會由障礙物變為可以走的空地，或從可以走的空地變成一個不能走的障礙物。此外，在魔幻迷宮中每次移動只能往下、往左或往右，並且走過的格子不能重複經過。

在殿壬找貓貓的過程中，貓貓因為等著等著太無聊了，居然在計算殿壬從起始點出發，找到貓貓的方法數有幾種！

現在，給你過程中迷宮所發生的 Q 個事件。請你依照事件，告訴貓貓殿壬有多少方法可以找到他呢？

Input

輸入第一行有三個整數 N, M, Q，代表迷宮可以用一個 $N \times M$ 的方格圖表示，並且在殿壬找貓貓的過程中，魔幻迷宮發生了 Q 個事件。

接下來的 N 行，每行有一個長度為 M 的 $0/1$ 字串。若第 i 行、第 j 列中的字元為 0，代表 (i, j) 方格是個空地；若第 i 行、第 j 列的字元為 1，代表 (i, j) 方格有障礙物。

接下來的 Q 行為殿壬找貓貓的過程中，魔幻迷宮所發生的事件，每個事件可以用三個整數來表示，且形式為以下兩種中的一種：

- 1 x y: 代表 (x, y) 格子的狀態改變了，也就是由空地變為障礙物、或由障礙物變為空地。

- 2 a b: 代表貓貓想知道在當前的迷宮狀態之下，殿壬從 $(1, a)$ 出發，走到貓貓在的位置 (N, b) 的方法數除以 $10^9 + 7$ 的餘數。若 $(1, a)$ 或 (N, b) 是障礙物，視同方法數為 0。

- $1 \leq N, Q \leq 50000$
- $1 \leq M \leq 10$
- $1 \leq x \leq N$
- $1 \leq y \leq M$
- $1 \leq a, b \leq M$

Output

對於每個形式為 2 a b 的事件，請輸出一行包含一個整數，代表殿壬找到貓貓的方法數除以 $10^9 + 7$ 的餘數。

Sample Input 1	Sample Output 1
2 2 3 00 00 2 1 2 1 1 2 2 1 2	2 1

C. 垃圾處理

Problem ID: garbage

你我生存在的這個世界上實在有太多垃圾了，同樣身為垃圾的 bb 決定要來親手改善這個問題。現在有 N 個垃圾被由左至右擺放在一個筆直的垃圾處理場中，每個垃圾都對應到一個正整數 t_i，代表這個垃圾的種類。bb 可以執行下方的操作任意多次：

- 壓縮：把其中兩個**相鄰且種類相同**的垃圾壓縮成一個新的垃圾（壓縮不同種類的垃圾有可能會造成爆炸，所以這是不被允許的），這個新的垃圾的種類與原本那兩個垃圾的種類相同。例如，原本有 4 個垃圾，由左至右的種類是 1 2 2 1，使用「壓縮」的操作可以把中間兩個垃圾壓縮成一個，剩下 3 個由左至右種類是 1 2 1 的垃圾。

現在 bb 想要知道，他是否能夠透過**不限次數**的壓縮操作，把原本的 N 個垃圾壓縮成**只剩下一個**。

Input

輸入的第一行有一個正整數 N，代表垃圾的數量。

第二行有 N 個以空格分開的正整數 $t_1, t_2, \ldots, t_N$，代表由左至右每個垃圾的種類。

- $1 \leq N \leq 10^5$
- $1 \leq t_i \leq 10^5$

Output

輸出只有一行。如果 bb 可以在**不限次數**的壓縮操作後將所有垃圾**壓縮成一個**，請輸出 "Yes"（不含雙引號），否則請輸出 "No"（不含雙引號）。

Sample Input 1	Sample Output 1
2 1 1	Yes

Sample Input 2	Sample Output 2
3 1 2 1	No

Sample Input 3	Sample Output 3
1 5	Yes

D. 分裁判問題

Problem ID: partition

傳說中，累積參加 20 次 NPSC (National Problem Solving Contest)，就會獲得最終大賽 **NPSC**(National Problem Setting Contest) 的門票！

但是 **NPSC** 每 20 年才會舉辦一次，而且每次僅能有 N 位參賽者。參加 **NPSC** 這個最高殿堂是每個 NPSC 裁判的心願（也只有身為 NPSC 裁判，才有可能參加 20 年 NPSC）。正巧，今年剛好有 $2N$ 個裁判累積參加滿 20 次 NPSC ，獲得了參加 **NPSC** 的資格。

身為 **NPSC** 的主辦，你希望最終來參加的是萬中選一的選手，因此，你決定將這 $2N$ 位符合資格的 NPSC 裁判分為兩隊，分別有 N 位裁判。為了使這 $2N$ 位裁判能使出渾身解術來競爭這 20 年一次的 **NPSC** 參賽權，你事先調查了這 $2N$ 個裁判兩兩之間的競爭指數，你希望分成兩隊之後，在不同隊伍間的裁判兩兩競爭指數的和能越大越好。但光是規劃這 20 年一次的 **NPSC**，就已經讓你傷透腦筋，因此，你決定寫個程式來幫你找出最大可能的競爭指數和。

Input

輸入第一行，包含一個正整數 N，代表有 N 位裁判能獲得 **NPSC** 的參賽權。接下來 $2N$ 行，每行包含 $2N$ 個以空格隔開的非負整數 v_{ij}，代表第 i 位裁判與第 j 位裁判間的競爭指數。

- $1 \leq N \leq 14$
- $0 \leq v_{ij} \leq 10^9$
- $v_{ij} = v_{ji}$
- $v_{ii} = 0$

Output

輸出一行包含一個整數，代表最大可能的競爭指數總和。

Note

兩兩裁判間競爭指數只需要算一次。即裁判 i 與裁判 j 若在不同隊，只需要將 v_{ij} 算入總和之中，不需要再加上 v_{ji}。

Sample Input 1	Sample Output 1
1 0 3 3 0	3

E. 選裁判問題

Problem ID: subset

身為 NPSC (National Program Squeezing Contest) 大賽的裁判長，最困難的其實不是把有趣且新奇的題目出出來，而是要從為數不多的裁判候選人中選出最適合的裁判群。

目前共有 N 位符合資格的裁判候選人，各各身懷絕技，能夠出各種讓參賽者會心一笑又不落俗套的有趣題目。但是，在這個小圈子中，裁判非常容易會認識參賽者，也許是學長姊跟學弟妹，也許是往日同隊的隊友，也可能是師徒關係。為了使 NPSC 成為一個公平、公正、公開的比賽，裁判長必須盡力的避嫌。

因此，裁判長調查了這 N 位符合資格的裁判候選人，以及 M 位當屆 NPSC 報名的參賽者，將每位參賽者與裁判是否認識的關係蒐集起來。現在，裁判長決定組織一個恰好有 K 位裁判的裁判團，裁判長想知道在所有可能的選擇當中，完全不與這 K 位裁判相識的參賽者最多有幾位。也就是說，對於所有選定 K 位裁判的方案中，與選定的 K 位裁判皆不相識的參賽者總人數最多有幾人？

由於裁判長仍在考慮最終題目的數量，因此他想知道對於所有可能的 K，最多能有多少滿足條件的參賽者。

Input

輸入第一行，包含兩個以空格隔開的正整數 N, M，分別代表候選裁判人數，以及報名參加的參賽者總數。接下來 N 行，每行包含一個長度為 M 的 01 字串，若第 i 行的第 j 個字元為 `'1'`，則代表第 i 位裁判與第 j 位參賽者相識；若第 i 行的第 j 個字元為 `'0'`，則代表第 i 位裁判與第 j 位參賽者不相識。

- $1 \leq N \leq 20$
- $1 \leq M \leq 10^6$

Output

請輸出 N 行，第一行包含一個整數代表 $K = 1$ 時，最多能有多少位參賽者、第二行代表 $K = 2$ 時最多的參賽者數、...、第 N 行代表 $K = N$ 時最多的參賽者數。

Sample Input 1	Sample Output 1
2 3 110 010	2 1

F. 地底探險

Problem ID: underground

U1 時空的小 B 是一位地底世界的探險家，他常常嗑了一堆ドーパミン（dopamine，多巴胺）後進入地底世界探險。然而小 B 有時候會因為不小心嗑了太多ドーパミン導致太嗨，讓他在地底世界中迷路，沒辦法好好在地底世界探險。於是小 B 決定告訴你他在地底世界探險的移動路徑以及他做了什麼事情，請在小 B 需要時告訴他他想要的資訊。

地底世界是由大量的地底空間及隧道構成的。每個隧道會連接兩個深度恰好差一的地底空間，其中深度的定義是從地表到達這個地底空間最少所需經過的隧道數量。每個地底空間可以被複數個隧道連接，但其中一定有唯一的一個隧道連到深度恰好為當前地底空間深度減一的空間。另外為了區別每個空間，每個空間都會被給予一個名字。小 B 在探險時可以做一些事情，例如挖一條新的隧道通往新的地底空間，使一條隧道連接著的地底空間崩塌，或是透過一個隧道走到某個相鄰的空間。簡單來說：

1. 小 B 可以挖一條隧道從目前所在的空間通往一個新的地底空間，並且小 B 會順便為此新地底空間命名。為了能夠唯一的分辨每個地底空間，同個空間連接到的深度為當前空間深度加一的地底空間都有不同的名字。
2. 小 B 可以讓一條從目前所在的空間通往深度較深的地底空間的隧道崩塌。而在此隧道崩塌後，必須通過這個隧道才能抵達的所有深度較深的地底空間都會一起崩塌。
3. 小 B 可以從目前所在的空間經過恰好一條隧道後，走到某個相鄰的空間。

如同前文所說，因為小 B 太嗨了，所以有時候小 B 會忘記他在哪個空間，或是忘記從現在所在的空間經過恰好一條隧道後能通往哪些深度較深的地底空間，又或是做出一些其他的危險事項。一旦發生了這種情況，請立刻告訴小 B 他該知道的事情。

小 B 會依序告訴你他要做的動作或是想知道的事情，每件事情的格式如下所述：

1. `"dig meow"`：代表小 B 從當前空間挖了一條新的隧道，且將此隧道通往的新地底空間命名為 `meow`。如果已經存在能經過恰好一個隧道前往且深度較深的地底空間擁有相同的名字，請輸出`"'meow' exist!"`，並無視此條資料，否則不需要輸出。請自行將 `meow` 替換成該資料記載的名字。
2. `"collapse meow"`：代表小 B 讓連接著目前空間與深度較深且命名為 `meow` 的地底空間的隧道崩塌。如果不存在這樣的地底空間，請輸出`"'meow' not exist!"`，並無視此條資料，否則不需要輸出。請自行將 `meow` 替換成該資料記載的名字。

3. "go to meow"：代表小 B 前往了某個恰透過一條隧道連接著，且深度比當前空間深的，且名稱為 meow 的地底空間。如果不存在這樣的地底空間，請輸出"'meow' not exist!"，並無視此條資料，否則不需要輸出。請自行將 meow 替換成該資料記載的名字。

4. "go back"：代表小 B 前往了那個唯一的，恰透過一條隧道連接著，且深度比當前空間淺的空間。如果小 B 現在已經在地表了的話，請輸出"You are on the ground!"，並無視此筆資料，否則不需要輸出。

5. "where am I"：代表小 B 忘記了自己目前在哪裡。請列出從地表開始，到達這個空間之前的所有空間的名字，再輸出當前空間的名字。如果要被印出的名字數量太多，則只需要印出最後 100 個即可。詳細格式請參考範例測試資料及 Note 部份。

6. "where can I go"：代表小 B 忘記了有哪些地底空間透過恰一條隧道連接著當前空間，且深度比當前空間深。請依照字典序輸出所有符合條件的地底空間的名字，並用恰一個空格分隔那些名字。如果符合條件的地底空間的名字超過 100 個，則只需輸出字典序前 100 小的地底空間的名字，並加上三個小數點。詳細請參考範例測試資料及 Note 部份。請不要輸出多餘的空格。

以上格式在輸入時都不包含雙引號。

Input

輸入的第一行是一個正整數 N，代表小 B 給你了 N 筆紀錄。接下來的 N 行，每一行恰為一筆紀錄，格式如上所述。

- $1 \leq N \leq 50000$
- $1 \leq$ 每個地底空間的名字的長度 ≤ 10
- 每個地底空間的名字只包含小寫英文字母

Output

請閱讀題目敘述，輸出該輸出的東西。

由於輸出量可能會非常大，所以請將要被輸出的內容**全部**送到下面提供的函式 `AddAnswer`，並在最後呼叫 `PrintAnswer` 用來輸出答案。

除了 `PrintAnswer` 輸出的內容之外，不建議自行輸出任何東西，以免造成預期外的結果。

```
int HashValue = 01234567;
void AddAnswer(char *s) {
    int n = strlen(s);
    for (int i = 0; i < n; ++i)
        HashValue = (HashValue * 131ll + s[i]) % 1000000123;
}
void PrintAnswer() {
    printf("%d\n", HashValue);
}
```

Note

1. `"where am I"` 的輸出說明：完整輸出的範例請參考下方：

   ```
   ground -> firstname -> secondname -> thirdname -> meow -> pog
   ```

   請注意，第一項一定會是 `ground`。

   只需要輸出後四個的輸出範例：

   ```
   -> secondname -> thirdname -> meow -> pog
   ```

   請留意若有名字被省略時，前面有 `->` 要輸出。

2. "where can I go" 的輸出說明：完整輸出的範例請參考下方：

```
a aa aaa aaaa bb bcd meow zoo zzz
```

只需要輸出前五個的輸出範例：

```
a aa aaa aaaa bb ...
```

請注意在最後一個名字後方有接著`...`。`...` 只有在有名字被省略時才需要輸出，例如總共只有五個名字，則不需要加上`...`。

另外，如果沒有任何符合條件的地底空間，也要輸出一個換行字元，詳細請參考下方的範例。

3. 以上兩點當中，實際上要輸出的數量請參考題目敘述。

範例測試資料輸出的字串原本如下：

```
ground

'd' exist!
ground -> a -> a -> a
a aa b c d
'e' not exist!
'a' not exist!

'a' not exist!
You are on the ground!
```

Sample Input 1	Sample Output 1
27 where am I where can I go dig a dig b dig c dig d dig d go to a dig a go to a dig a go to a where am I go back go back go back dig aa where can I go collapse e collapse a go to a dig a go to a where can I go go to a go back go back	603303985

This page is intentionally left blank.

G. 盤子分類

Problem ID: dishes

小 Y 是個喜歡收集盤子的收藏家，他在家中收集了各式各樣的盤子，並且按照盤子的顏色分類，把每種顏色的盤子疊成一疊。

有一天，他發現他收集的盤子竟然被打亂了！原來是小 P 一時無聊，就把黃色和白色這兩疊盤子隨意交換。小 Y 無法忍受黃白夾雜的盤子疊成一疊，因此他決定要趕快把這兩種顏色的盤子分類放好（一疊全部都是黃色，一疊全部都是白色。至於哪一疊是黃色、哪一疊是白色對小 Y 來講並不重要）。

不幸的是，由於小 Y 家實在有太多盤子了，所以他找不到多餘的空間可以整理這些盤子（因為一不小心可能會摔碎其他的盤子）。因此，他決定直接用這兩疊盤子的空間來把盤子分類。具體來說，小 Y 每次會選一個盤子，把它抽出來後並放到另一疊盤子的最上面，一直重複這樣的動作直到盤子分類好為止。但是要把盤子從一整疊盤子的中間抽出來就代表要暫時把上面所有的盤子抬起來，因此如果那個要抽出來的盤子上面有 x 個盤子，那抽出這個盤子會讓小 Y 耗費 x 單位的體力。而整個小 Y 分類盤子的過程所耗費的體力就是每一次動作所耗費的體力的加總。

小 Y 想要用最少的體力把盤子分類放好。但是他的數學不好，不太會解最優化問題。所以請你幫忙他計算出他最少需要耗費多少單位的體力才能把盤子分類放好。

Input

輸入只有兩行，兩行各有一個僅以 `Y` 和 `W` 組成的字串，分別代表兩疊盤子中**由上而下**每個盤子的顏色，`Y` 代表黃色、`W` 代表白色。

- 兩疊盤子的盤子數量均介於 1 和 20000 之間（含）。

Output

請輸出一行包含一個整數，代表小 Y 最少需要耗費多少單位的體力才能把盤子分類放好。

Sample Input 1	Sample Output 1
YYYYYY YWWWWW	0

Sample Input 2	Sample Output 2
YW WY	1

Sample Input 3	Sample Output 3
WW YWWYW	4